

# WB MGW WIFI RS485 Converter

---

## Преобразователь интерфейсов WB-MGW Wi-Fi — RS-485

### Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке:  
[https://wireboard.com/wiki/WB\\_MGW\\_WIFI\\_RS485\\_Converter](https://wireboard.com/wiki/WB_MGW_WIFI_RS485_Converter)

Этот документ составлен автоматически из основной страницы документации  
и ссылок первого уровня.

# Содержание

[WB MGW WIFI RS485 Converter](#)

[Протокол Modbus](#)

[Test](#)

## WB MGW WIFI RS485 Converter

---

Статья в процессе наполнения и редактирования (кандидат на удаление)

### Contents

---

[Назначение](#)

[Технические характеристики](#)

[Обмен данными](#)

[Режимы работы](#)

[Монтаж и управление](#)

[Представление в веб-интерфейсе](#)

[Управление по Modbus](#)



Преобразователь интерфейсов WB MGW  
Wi-Fi — RS-485

### Назначение

---

Преобразователь интерфейсов WB-MGW Wi-Fi — RS-485 предназначен для создания моста между сетями Wi-Fi и RS-485. Преобразователь может служить как для прозрачного подключения удаленных устройств Modbus RS-485 к интерфейсу RS-485, так и для создания моста RS-485 — Wi-Fi — RS-485 для Modbus-устройств.

### Технические характеристики

---

Параметр	Значение
<b>Питание</b>	
Напряжение питания интерфейсной части	9 В — 24 В постоянного тока
Потребляемая мощность	0,? Вт
<b>Каналы управления нагрузкой</b>	
<b>Коммуникация</b>	
Протокол обмена данными	Modbus RTU, TCP/IP, Wi-Fi
Параметры интерфейса RS-485	Задаются программно, по умолчанию: скорость 9600 бит/с; данные — 8 бит; четность N; стоп-биты 2
Параметры интерфейса Wi-Fi	Задаются программно, по умолчанию
<b>Габариты</b>	
Габариты	45 x 35 x 14 мм
<b>Условия эксплуатации</b>	
Температура воздуха	От -40°C до +80°C
Относительная влажность воздуха	До 92%, без конденсации влаги

## Обмен данными

---

На физическом уровне подключается через интерфейс RS-485. Подробнее смотрите страницу [Протокол Modbus](#). Подробности смотрите в разделе [Управление по Modbus](#).

## Режимы работы

---

Преобразователь WB-MGW представляет собой конвертер, прозрачно преобразующий сигналы Modbus RTU шины RS-485 в пакеты TCP/IP и обратно. В режиме Wi-Fi преобразователь может выступать в роли точки доступа, в роли беспроводного клиента или одновременно работать как точка доступа и беспроводной клиент. В отличие от других Modbus-устройств, подключаемых к линии RS-485, конвертер не имеет Modbus-адреса и прозрачно передает и принимает сигналы шины RS-485. При этом как приемник и передатчик устройство подчиняется всем требованиям протокола Modbus RTU об очередности передачи данных по линии.

## Монтаж и управление

---

## Представление в веб-интерфейсе

---

# управление по Modbus

---

Test

# Протокол Modbus

- [English](#)
- русский

## Contents

---

### [Основные понятия](#)

### [Структуры данных Modbus](#)

### [Модель данных Modbus](#)

### [Адреса регистров](#)

[Нестандартная адресация](#)

[Пример описания регистров в документации](#)

### [Коды функций чтения и записи регистров](#)

### [Формат данных запросов и ответов Modbus](#)

### [Коды исключений \(ошибки\) Modbus](#)

### [Вычисление контрольной суммы Modbus](#)

## Основные понятия

Modbus - это протокол прикладного (седьмого) уровня модели OSI. Чаще всего он служит для обмена данными между устройствами автоматизации и реализован в виде "протокола ответов на запросы (request-reply protocol)".

В устройствах Wiren Board данные Modbus передаются по последовательным линиям связи RS-485. В последовательных линиях связи протокол RS-485 полудуплексный и работает по принципу «клиент-сервер». Каждое устройство в сети (кроме ведущего см. далее) имеет адрес от 1 до 247, адрес 0 используется для широковещательной передачи данных всем устройствам, а адреса 248-255 считаются зарезервированными согласно спецификации Modbus, их использование не рекомендуется.

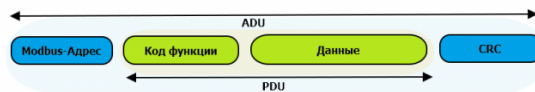
Существует две спецификации протокола: Modbus RTU и Modbus ASCII. В Modbus RTU передается 11-битный символ, состоящий из 1 стартового бита, 8 бит данных (начиная с младшего бита), бит четности (необязателен) и 2 стоповых бита - если бит четности не передается, или 1 стоповый бит - если бит четности передается. Такой символ передает 1 байт данных. В устройствах Wiren Board по умолчанию бит контроля четности не передается и используется 2 стоповых бита. В Modbus ASCII каждый байт передается двумя символами, представляющими ASCII-коды младшей и старшей четырехбитной группы байта (пример). Modbus RTU передает

больше информации при той же скорости последовательной линии, и в устройствах Wiren Board используется именно он. Все дальнейшее описание относится к Modbus RTU.

Ведущее устройство ("мастер", или "клиент") периодически опрашивает "ведомое", или "сервер". Ведущее устройство не имеет адреса, передача сообщений от устройства-сервера ведущему без запроса ведущего в протоколе не предусмотрена.

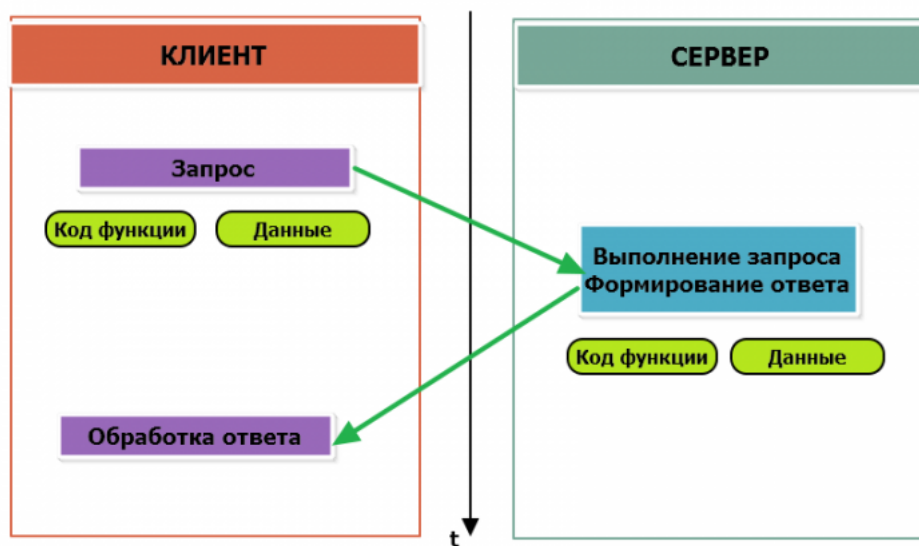
Пакет данных Modbus выглядит, как это показано на рисунке. **PDU** (Protocol Data Unit) — общая часть пакета MODBUS, включающая код функции и данные пакета. **ADU** (Application Data Unit) — полный пакет MODBUS. Включает в себя специфичную для физического уровня часть пакета и PDU.

Для последовательных линий в заголовке ADU передается адрес устройства, а в конце — контрольная сумма CRC16. Максимальный размер ADU в последовательных коммуникационных линиях составляет **253 байта** (из максимальных, разрешенных спецификацией 256 байт вычитается 1 байт адреса и два байта контрольной суммы). Для справки — в Modbus TCP максимальная длина пакета составляет 260 байт.

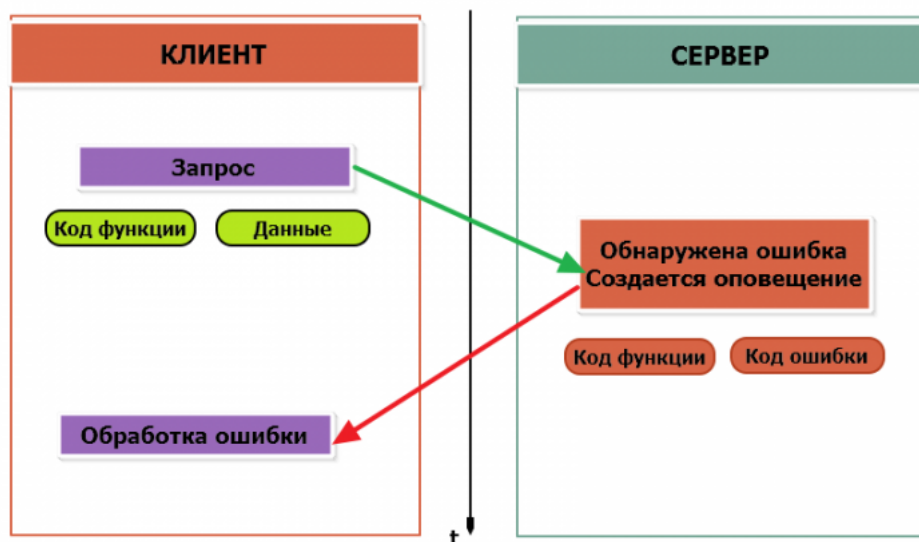


Датаграмма Modbus в общем виде

Функция кодируется одним байтом и определяет, какое действие должно выполнить устройство-сервер. Значение кодов функций лежат в диапазоне от 1 до 255, причем коды от 128 до 255 зарезервированы для сообщений об ошибках со стороны устройства-сервера. Код 0 не используется. Размер блока данных может варьироваться от нуля до максимально допустимого. Если обработка запроса прошла без ошибок, то устройство-сервер возвращает пакет ADU, содержащий запрошенные данные.



Modbus-транзакция, прошедшая без ошибок



Modbus-транзакция с ошибками

При возникновении ошибки устройством возвращается код ошибки. При обычной транзакции код функции в ответе возвращается без изменений; при ошибке старший бит кода функции устанавливается в единицу (то есть *код функции* + 0x80). Так же есть таймаут ожидания ответа от ведомого устройства — бессмысленно долго ждать ответ, который, возможно, никогда и не придет.

## Структуры данных Modbus

В Modbus принято кодировать адреса и данные в формате big-endian, то есть в формате, когда байты следуют, начиная со старшего: например, при передаче шестнадцатеричного числа 0x1234 сначала устройством будет принят байт 0x12, а затем — 0x34. Для передачи данных другого типа, например, чисел с плавающей запятой (float), текстовых строк, даты и времени суток и т.п. производитель может выбрать свой собственный способ кодирования — для расшифровки получаемых данных важно ознакомиться со спецификацией производителя устройства.

## Модель данных Modbus

Обмен данными с Modbus-устройствами происходит через регистры. В протоколе Modbus определяется четыре типа регистров, показанных в таблице:

Таблица	Размер	Доступ
<b>Регистры флагов (Coils)</b>	1 бит	чтение и запись
<b>Дискретные входы (Discrete Inputs)</b>	1 бит	только чтение
<b>Регистры хранения (Holding Registers)</b>	16-битное слово	чтение и запись
<b>Регистры ввода (Input Registers)</b>	16-битное слово	только чтение

**Регистры флагов (Coils)** хранят однобитные значения - то есть могут находиться в

состоянии 0 или 1. Такие регистры могут обозначать текущее состояние выхода (включено реле). Название "coil" буквально и означает обмотку-актюатор электромеханического реле. Регистры флагов допускают как чтение, так и запись.

**Дискретные входы** (Discrete Inputs) также являются однобитными регистрами, описывающими состояние входа устройства (например, подано напряжение — 1). Эти регистры поддерживают только чтение.

**Регистры хранения** (Holding Registers) и **регистры ввода** (Input Registers) представлены двухбайтовым словом и могут хранить значения от 0 до 65535 (0x0000 — 0xFFFF). Регистры ввода допускают только чтение (например, текущее значение температуры). Регистры хранения поддерживают как чтение, так и запись (для хранения настроек). В настоящее время во многих устройствах, в частности в устройствах Wigen Board, эти регистры не разделяются. Команды на чтение регистра хранения N и регистра ввода N обратятся к одному и тому же значению в адресном пространстве устройства.

## Адреса регистров

Регистры в стандарте Modbus адресуются с помощью 16-битных адресов. Адресация начинается с нуля. Адрес регистра, таким образом, может принимать значения от 0 до 65535.

Адресные пространства регистров, также называемые таблицами или блоками, могут быть различны для всех четырёх типов регистров. Это значит, что значения регистров с одинаковым адресом, но разным типом, в общем случае разные.

Например, при чтении регистра флагов (coil) номер 42, регистра дискретного входа (Discrete), регистров ввода и хранения (Input и Holding) с теми же адресами, можно получить четыре разных значения.

## Нестандартная адресация

В документации на некоторые, особенно старые, устройства адреса элементов (регистров) указываются в формате, не соответствующем стандарту. В этом формате тип элемента кодируется первой цифрой адреса, а адресация начинается не с нуля.

Например, регистр хранения с адресом 0 может записываться как 40001 или 400001, а Coil с адресом 0 как 000001.

В таблице перевода адресов в стандартный формат показаны диапазоны для двух разных нестандартных типов указания адресов и соответствующие им типы данных и диапазоны стандартных адресов.



Тип данных	Стандартные адреса	Стандартные адреса (hex)	Нестандартные адреса (5 цифр)	Нестандартные адреса (6 цифр)
<b>Флагов (Coils)</b>	0-65535	0x0000 - 0xFFFF	00001 - 09999	000001 - 065536
<b>Дискретных входов (Discrete)</b>	0-65535	0x0000 - 0xFFFF	10001 - 19999	100001 - 165536
<b>Регистры входов (Input Registers)</b>	0-65535	0x0000 - 0xFFFF	30001 - 39999	300001 - 365536
<b>Регистры хранения (Holding Registers)</b>	0-65535	0x0000 - 0xFFFF	40001 - 49999	400001 - 465536

Признаки использования нестандартной адресации:

- Адреса записываются в десятичном формате
- Во всех адресах пять или шесть цифр
- Адреса с недискретными данными (показания датчиков и т.п.) начинаются на 30 или 40

Часто рядом с нестандартными адресами указываются и адреса соответствующие стандарту, обычно в шестнадцатеричном формате. Стоит отметить, что физически в пакете данных передаются адреса в стандартном формате, независимо от способа представления их в документации.

### Пример описания регистров в документации

В готовых шаблонах устройств для контроллера Wiren Board есть шаблон для однофазного счетчика электроэнергии SDM220 (/usr/share/wb-mqtt-serial/templates/config-sdm220.json). В документации от производителя "Eastron SDM 220 Modbus Smart Meter Modbus Protocol Implementation V1.0" перечислены регистры и соответствующие им измеряемые параметры, например:

Address (Register)	Description	Units	Modbus Protocol Start Address Hex (Hi Byte Lo Byte)
<b>30001</b>	Line to neutral volts.	Volts	00 00
<b>30007</b>	Current.	Amps.	00 06
<b>30013</b>	Active power	Whatts	00 0C
<b>30019</b>	Apparent power	VoltAmps	00 12
...	...	...	... ..

Производитель в таблице приводит и логические, и физические адреса регистров, что позволяет нам с легкостью создать шаблон устройства и проиллюстрировать связь между логическими и физическими адресами Modbus-регистров.

```

"channels" : [
  {
    "name" : "Voltage",
    "type" : "voltage",
    "reg_type" : "input",
    "address" : "0x00",
    "format" : "float"
  },
  {
    "name" : "Current",
    "type" : "current",
    "reg_type" : "input",
    "address" : "0x06",
    "format" : "float"
  },
  {
    "name" : "Active Power",
    "type" : "power",
    "reg_type" : "input",
    "address" : "0x0c",
    "format" : "float"
  },
  {
    "name" : "Apparent Power",
    "type" : "power",
    "reg_type" : "input",
    "address" : "0x12",
    "format" : "float"
  }
],

```

Фрагмент шаблона счетчика SDM220

## Коды функций чтения и записи регистров

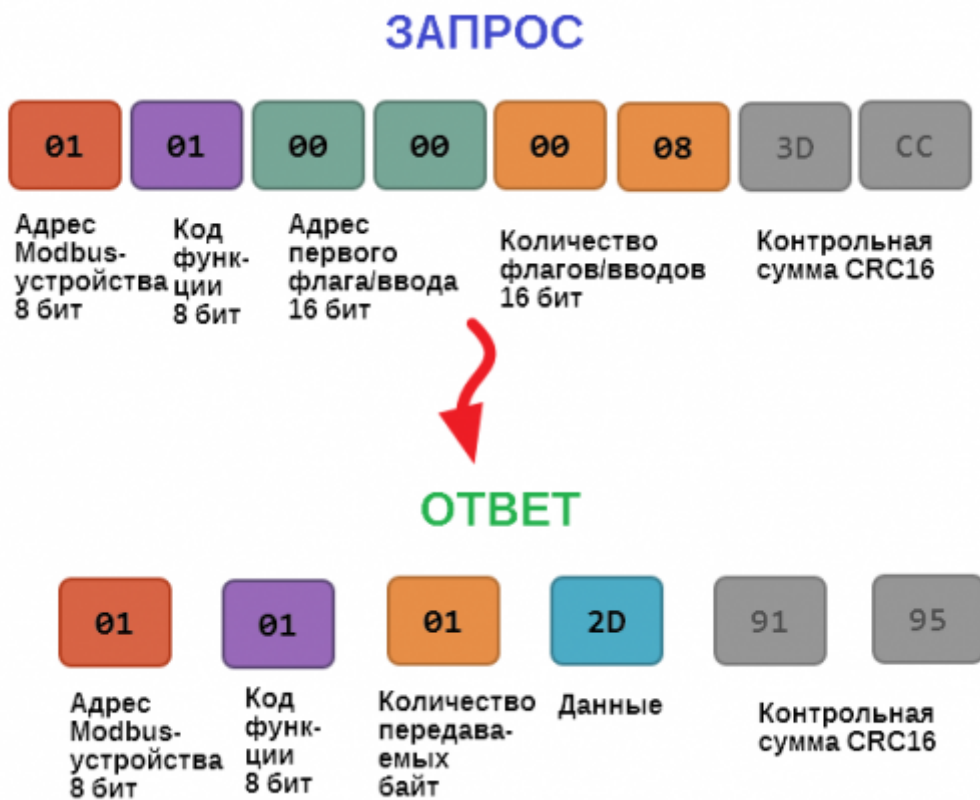
В следующей таблице приведены наиболее распространенные коды функций Modbus:

Код функции	HEX	Название	Действие
<b>1</b>	0x01	Read Coils	Чтение значений нескольких регистров флагов
<b>2</b>	0x02	Read Discrete Inputs	Чтение значений нескольких дискретных входов
<b>3</b>	0x03	Read Holding Registers	Чтение значений нескольких регистров хранения
<b>4</b>	0x04	Read Input Registers	Чтение значений нескольких регистров ввода
<b>5</b>	0x05	Write Single Coil	Запись одного регистра флагов
<b>6</b>	0x06	Write Single Register	Запись одного регистра хранения
<b>15</b>	0x0F	Write Multiple Coils	Запись нескольких регистров флагов
<b>16</b>	0x10	Write Multiple Register	Запись нескольких регистров хранения

Команды условно можно разделить по типам: чтение значений — запись значений; операция с одним значением — операция с несколькими значениями.

## Формат данных запросов и ответов Modbus

Рассмотрим подробнее, как происходит обмен данными между устройством-клиентом, отправляющим запрос, и устройством-сервером, отвечающим ему. На следующем рисунке показан обмен данными контроллера с устройством с адресом 0x01. Мы хотим прочесть 8 coil-регистров, начиная с первого.



Обмен данными в Modbus

В качестве данных мы получили шестнадцатеричное число 0x2D, то есть состояние восьми coil-регистров в двоичном виде такое: 0b10110100.

В следующей таблице приведены структуры данных запросов и ответов для основных функций Modbus.

Код функции	Запрос	Ответ
<b>1 (Read Coils) и 2 (Read Discrete Inputs)</b>	<ul style="list-style-type: none"> <li>▪ Адрес первого регистра флагов или входного регистра (16 бит)</li> <li>▪ Количество данных (8 значений на байт) (16 бит)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Число передаваемых байт (8 бит)</li> <li>▪ Значения регистров флагов или входных регистров (8 значений на байт)</li> </ul>
<b>3 (Read Holding Registers) и 4 (Read Input Registers)</b>	<ul style="list-style-type: none"> <li>▪ Адрес первого регистра (16 бит)</li> <li>▪ Количество регистров, которые нужно прочесть</li> </ul>	<ul style="list-style-type: none"> <li>▪ Число передаваемых байт (8 бит)</li> <li>▪ Значения регистров (16 бит на 1 регистр)</li> </ul>
<b>5 (Write Single Coil)</b>	<ul style="list-style-type: none"> <li>▪ Адрес регистра (16 бит)</li> <li>▪ Значение, которое нужно записать (0 — выключить, 0xFF00 — включить)</li> </ul>	Ответ аналогичен запросу
<b>6 (Write Single Register)</b>	<ul style="list-style-type: none"> <li>▪ Адрес регистра(16 бит)</li> <li>▪ Новое значение регистра (16 бит)</li> </ul>	Ответ аналогичен запросу
<b>15 (Write Multiple Coils)</b>	<ul style="list-style-type: none"> <li>▪ Адрес первого регистра флагов для записи (16 бит)</li> <li>▪ Количество регистров флагов для записи (16 бит)</li> <li>▪ Количество передаваемых байт данных для регистров флагов (8 бит)</li> <li>▪ Данные (8 регистров флагов на байт)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Адрес первого coil-регистра (16 бит)</li> <li>▪ Количество записанных coil-регистров(16 бит)</li> </ul>
<b>16 (Write Multiple register )</b>	<ul style="list-style-type: none"> <li>▪ Адрес первого регистра хранения для записи (16 бит)</li> <li>▪ Количество регистров хранения для записи (16 бит)</li> <li>▪ Количество передаваемых байт данных для регистров (8 бит)</li> <li>▪ Данные (16 байт на регистр)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Адрес первого регистра хранения (16 бит)</li> <li>▪ Количество записанных регистров хранения(16 бит)</li> </ul>

## Коды исключений (ошибки) Modbus

Если запрос не может по той или иной причине быть обработан устройством-сервером, то в ответ он отправляет сообщение об ошибке. Сообщение об ошибке содержит адрес Modbus-устройства, код функции, при выполнении которой произошла ошибка, увеличенный на 0x80, код ошибки и контрольную сумму:

## ОШИБОЧНЫЙ ЗАПРОС



## СООБЩЕНИЕ ОБ ОШИБКЕ



Транзакция завершилась с ошибкой

В этом случае мы попытались обратиться к несуществующему адресу регистра 0xFFFF и попытались прочесть 8 регистров флагов. В результате мы получили код ошибки 0x03 — "В поле данных передано неверное значение".

Наиболее распространенные коды ошибок Modbus приведены в следующей таблице:

Код ошибки	Название ошибки	Что означает
1	Illegal Function	В запросе был передан недопустимый код функции
2	Illegal Data Address	Указанный в запросе адрес не существует
3	Illegal Data Value	Неверный формат запроса, например количество байт в запросе не соответствует ожидаемому.  <b>Примечание:</b> несмотря на название, эта ошибка не говорит о том, что само значение регистра неправильное или ошибочное, и должна использоваться только для ошибок формата запроса.
4	Server Device Failure	Произошла невосстановимая ошибка на устройстве при выполнении запрошенной операции
5	Acknowledge	Запрос принят, выполняется, но выполнение потребует много времени; необходимо увеличить таймаут.
6	Server Device Busy	Устройство занято обработкой предыдущего запроса.
7	Negative Acknowledge	Устройство не может выполнить запрос, необходимо получить от устройства дополнительную диагностическую информацию. Возможно, требуется тех. обслуживание.
8	Memory Parity Error	Ошибка четности при обращении к внутренней памяти устройства.

## Вычисление контрольной суммы Modbus

Для протокола Modbus RTU 16-битная контрольная сумма (CRC) вычисляется по алгоритму, описанному в спецификации Modbus, в документе "Modbus Serial Line Protocol and Implementation Guide", раздел "CRC-generation". Передающее устройство формирует два байта контрольной суммы на основе данных сообщения, а принимающее устройство заново вычисляет контрольную сумму и сравнивает с полученной. Совпадение принятой и вычисленной контрольной суммы Modbus RTU считается индикатором успешного обмена данными.

В случае ограниченных вычислительных ресурсов для вычисления контрольной суммы существует функция, использующая табличные значения (также приведена в спецификации).

# Test

Это черновик страницы. Последняя правка сделана 21.05.2022 пользователем A.Degtyarev.

## Contents

**Заголовок 1**

**Заголовок 2**

**Заголовок 3**

**Заголовок 4**

**Заголовок 5**

**Заголовок 6**

**Заголовок 7**

## Заголовок 1

---

Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста

Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста м

Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста м

## Заголовок 2

---

Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста







Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста Много текста Много текста Много текста  
Много текста Много текста Много текста м

---

Retrieved from "<https://wirenboard.com/wiki/Служебная:Print/>"

---

- Privacy policy
- About Wiren Board
- Disclaimers
-